LSTN: Latent Subspace Transfer Network for Unsupervised Domain Adaptation

Shanshan Wang and Lei Zhang*

College of Communication Engineering, Chongqing University,
No. 174 Shazheng street, Shapingba district, Chongqing 400044, China
{wangshanshan,leizhang}@cqu.edu.cn

Abstract. For handling cross-domain distribution mismatch, a specially designed subspace and reconstruction transfer functions bridging multiple domains for heterogeneous knowledge sharing are wanted. In this paper, we propose a novel reconstruction-based transfer learning method called Latent Subspace Transfer Network (LSTN). We embed features/pixels of source and target into reproducing kernel Hilbert space (RKHS), in which the high dimensional features are mapped to nonlinear latent subspace by feeding them into MLP network. This approach is very simple but effective by combining both advantages of subspace learning and neural network. The adaptation behaviors can be achieved in the method of joint learning a set of hierarchical nonlinear subspace representation and optimal reconstruction matrix simultaneously. Notably, as the latent subspace model is a MLP Network, the layers in it can be optimized directly to avoid a pre-trained model which needs large-scale data. Experiments demonstrate that our approach outperforms existing non-deep adaptation methods and exhibits classification performance comparable with that of modern deep adaptation methods.

Keywords: Domain Adaptation · Latent Subspace · MLP

1 Introduction

In computer vision, the dilemma of insufficient labeled data is common in visual big data, one of the prevailing problems in the practical application, is that when training data (source domain) exhibit a different distribution to test data (target domain), the task-specific classifier usually does not work well on related but distribution mismatched tasks.

Domain adaptation (DA) [5, 15, 31] techniques that are capable of easing such domain shift problem have received significant attention from engineering recently. It is thus of great practical importance to explore DA methods. These models allow machine learning methods to be self-adapted among multiple knowledge domains, that is, the trained model parameters from one data domain can be adapted to another domain. The assumption underlying DA is that, although the domains differ, there is sufficient commonality to support such adaptation.

* Corresponding author. The first author is a student author.

A substantial number of approaches to domain adaptation have been proposed in the context of both shallow learning and deep learning, which bridge the source and target domains by learning domain-invariant feature representations without using target labels, such that the classifier learned from the source domain can also be applied to the target domain.

Visual representations learned by deep CNNs are fairly domain-invariant. Relatively high accuracy is always reported over a lot of visual tasks using off the-shelf CNN representations [4, 26]. However, on one hand, deep neural networks which learn abstract feature representations can only reduce, but not remove, the cross-domain discrepancy. On the other, training a deep model relies on massive amounts of labeled data. Compared with deep method, shallow domain adaptation methods which are more suitable for small-scale data usually fail to reach the high accuracy as deep learning.

Our work is primarily motivated by [21] which investigates a provocative question that domain adaptation is necessary even if CNN-based features are powerful. We thus proposed a non-deep method which combines both advantages of subspace learning and neural networks inspired by [12]. Although this LSTN method is simple, it can achieve competitive results compared with deep methods. The main contribution and novelty of this work are threefold:

- In order to achieve the domain alignment, we propose a simple but effective net called Latent Subspace Transfer Network (LSTN). In order to get an optimal subspace representation, a joint learning mechanism is adopted for pursuing the latent subspace and reconstruction matrix simultaneously.
- The optimal latent subspace to map the source and target samples in LSTN is achieved by MLP network, which has a simple network structure but is effective. The model is a non-linear neural network and can be optimized directly to avoid a pre-trained model which needs large-scale data.
- In this simple network, we embed features/pixels of source and target into reproducing kernel Hilbert spaces (RKHS) as preprocessing before putting them into the optimization procedure. In this way ,the dimension of input and the cost of running time are both reduced.

2 Related Works

2.1 Shallow Domain Adaptation

A number of shallow learning methods have been proposed to tackle DA problems. Generally, these shallow domain adaptation methods comprise of three categories: Classifier based approaches, feature augmentation/transformation based approaches and feature reconstruction based approaches. [5] proposed an adaptive multiple kernel learning (AMKL) for web-consumer video event recognition. [35] proposed a robust domain classifier adaptation method (EDA) with manifold regularization for visual recognition. [19] also proposed a Transfer Joint Matching (TJM) which tends to learn a non-linear transformation across domains by minimizing the MMD based distribution discrepancy. [36] proposed a

Latent Sparse Domain Transfer (LSDT) method by jointly learning a subspace projection and sparse reconstruction across domains. Similarly, Shao et al. [25] proposed a LTSL method by pre-learning a subspace using PCA or LDA. Jhuo et al.[13] proposed a RDALR method, in which the source data is reconstructed by the target data using low-rank model. Recently, [21] proposed a LDADA method which can achieve the effect of DA without explicit adaptation by a LDA-inspired approach.

2.2 Deep Domain Adaptation

As deep CNNs become a mainstream technique, deep learning has witnessed a great achievements [29, 22, 32] in unsupervised DA. Very recently, [27] explored the performance improvements by combining the deep learning and DA methods.

Donahue et al. [4] proposed a deep transfer strategy for small-scale object recognition, by training a CNN network (AlexNet) on ImageNet. Tzeng et al. [29] proposed a CNN based DDC method which achieved successful knowledge transfer between domains and tasks. Long et al. [17] proposed a deep adaptation network (DAN) by imposing MMD loss on the high-level features across domains. Additionally, Long et al. [20] also proposed a residual transfer network (RTN) which tends to learn a residual classifier based on softmax loss. Hu et al. [12] proposed a non-CNN based deep transfer metric learning (DTML) method to learn a set of hierarchical nonlinear transformations for cross-domain visual recognition. Recently, GANs inspired adversarial domain adaptation methods have been preliminarily studied. Tzeng et al. proposed a novel ADDA method [30] for adversarial domain adaptation, in which CNN is used for adversarial discriminative feature learning. The work has shown the potential of adversarial learning in domain adaptation. In [11], Hoffman et al. proposed a CyCADA method which adapts representations at both the pixel-level and feature-level, enforcing cycle-consistency by leveraging a task loss.

However, most deep DA methods need large-scale data to train the model in advance, the insufficient data task involved is just used to fine tune the model. In contrast to these ideas, we show that one can achieve fairly good classification performance without pre-trained.

3 The Proposed latent subspace transfer network

3.1 Notation

In this paper, the source and target domain are defined by subscript "S" and "T". The training set of source and target domain are defined as $\boldsymbol{X}_S \in \mathbb{R}^{d\times n_S}$ and $\boldsymbol{X_T} \in \mathbb{R}^{d\times n_T}$, where d denotes dimension of input, n_S and n_T denote the number of samples in source and target domain, respectively. Let $\boldsymbol{C}$ represents the number of classes, $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n_S\times n_T}$ represents the reconstruction coefficient matrix. $\|\cdot\|_F$ and $\|\cdot\|_*$ denotes Frobenius norm and nuclear norm, respectively.

Notably, in order to reduce the feature dimension of input, we first embed the features in source and target domain into reproducing kernel Hilbert spaces

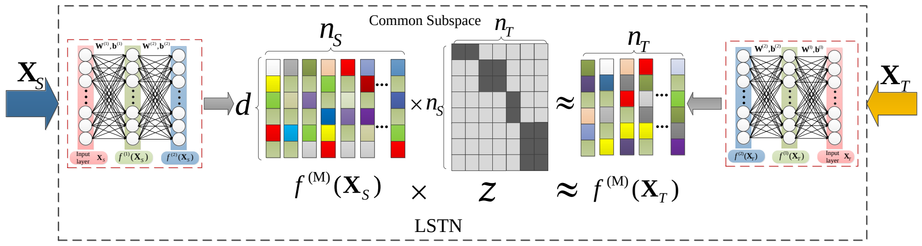


Fig. 1: The basic idea of the LSTN method is shown in Fig. 1. For each sample in the training sets from the source domain and the target domain, we pass it to the MLP network. We enforce the reconstruction constraint on the outputs of all training samples at the top of the network, in this way, the adaptation behaviors can be achieved by joint learning a set of hierarchical nonlinear subspace representation and optimal reconstruction matrix simultaneously. The network architecture of the MLP used in our methods is also shown in Fig. 1. The $\boldsymbol{X}$ is the data points in the input space, $f^{(1)}(\mathbf{X})$ is the output in the hidden layer and $f^{(2)}(\mathbf{X})$ is the resulting representation of $\boldsymbol{X}$ in the common subspace.In our experiments, the number of layers is set as $M = 2$.

(RKHSs) as preprocessing to get $\boldsymbol{X}_S$ and $\boldsymbol{X}_T$, then feed them to nonlinear neural network latent subspace. The kernel embedding represents a probability distribution $\mathbf{P}$ by an element in RKHS endowed by a kernel $k(\cdot)$ where the distribution is mapped to the expected feature map.

In our experiments, we consider a closer to reality case where no labeled training set is obtained from the target domain in unsupervised setting.

3.2 Model Formulation

As is shown is Fig. 1, unlike most previous transfer learning methods which usually seek a single linear subspace to map samples into a common latent subspace, we construct a multilayer perceptron network to compute the representations of each sample $\boldsymbol{x}$ by passing it to multiple layers of nonlinear transformations. By using such a network, besides the nonlinear mapping function can be explicitly obtained, the network structure is much simpler than deep methods which can avoid a pre-trained model.

Assume there are $M + 1$ layers in the designed network. The output $\boldsymbol{X}$ at the m^{th} layer is computed as:

$$f^{(m)}(\mathbf{X}) = \mathbf{h^{(m)}}=\varphi(\mathbf{Z^{(m)}}) = \varphi(\mathbf{W}^{(m)}\mathbf{h^{(m-1)}} + \mathbf{b}^{(m)}) \tag{1}$$

where $m = 1, 2, ..., M$ and $p^{(m)}$ units in the m^{th} layer. $\boldsymbol{W}^{(m)} \in \mathbb{R}^{p^{(m)} \times p^{(m-1)}}$ and $\boldsymbol{b}^{(m)} \in \mathbb{R}^{p^{(m)}}$ are the parameters of weight matrix and bias in this layer, the $\mathbf{Z}^{(m)} = \mathbf{W}^{(m)}\mathbf{h}^{(m-1)} + \mathbf{b}^{(m)}$ and $\varphi(\cdot)$ is a nonlinear activation function which operates component-wisely, such as widely used tanh or sigmoid functions. The nonlinear mapping $f^{(m)} : \mathbb{R}^{p^{(m-1)}} \rightarrow \mathbb{R}^{p^{(m)}}$ is a function in the m^{th} layer parameterized by $\{\mathbf{W}^{(i)}\}_{i=1}^{m}$ and $\{\mathbf{b}^{(i)}\}_{i=1}^{m}$. For the first layer, we assume $\mathbf{h}^{(0)} = \mathbf{X}$ ($\boldsymbol{X}_S$ or $\boldsymbol{X}_T$) and $p^{(0)} = d$.

For both source data $\boldsymbol{X}_S$ and target data $\boldsymbol{X}_T$, their probability distributions are different in the original feature space. In order to reduce the distribution difference, it is desirable to map the probability distribution of the source domain and that of the target domain into the common transformed subspace. The two domains are finally represented as $f^{(m)}(\boldsymbol{X}_S)$ and $f^{(m)}(\boldsymbol{X}_T)$ at the m^{th} layer of our designed network respectively, and their reconstruction error can be expressed by computing the squared Euclidean distance between the representations $f^{(M)}(\boldsymbol{X}_S)$ and $f^{(M)}(\boldsymbol{X}_T)$ at the last layer as:

$$D_{st}(\mathbf{X}_S, \mathbf{X}_T) = ||f^{(M)}(\mathbf{X}_S)\boldsymbol{\mathcal{Z}} - f^{(M)}(\mathbf{X}_T)||_F^2 \tag{2}$$

The low-rank representation is advantageous in getting the block diagonal solution for subspace segmentation, so that the global structure can be preserved. In constructing the reconstruction matrix $\boldsymbol{\mathcal{Z}}$ in this paper, the low-rank regularizer is used to better account for the global characteristics. By combining the reconstruction loss and the regularizer item together, the general objective function of the proposed LSTN model can be formulated as follows.

$$\begin{aligned} &\min_{f^{(m)},\boldsymbol{\mathcal{Z}}} J = D_{st}(\mathbf{X}_S, \mathbf{X}_T) + \lambda||\boldsymbol{\mathcal{Z}}||_* + \gamma \sum_{m=1}^{M} (||\mathbf{W}^{(m)}||_F^2 + ||\mathbf{b}^{(m)}||_2^2) \\ &= ||f^{(M)}(\mathbf{X}_S)\boldsymbol{\mathcal{Z}} - f^{(M)}(\mathbf{X}_T)||_F^2 + \lambda||\boldsymbol{\mathcal{Z}}||_* + \gamma \sum_{m=1}^{M} (||\mathbf{W}^{(m)}||_F^2 + ||\mathbf{b}^{(m)}||_2^2) \end{aligned} \tag{3}$$

where $\lambda(\lambda > 0)$ and $\gamma(\gamma > 0)$ are the tunable positive regularization parameters.

3.3 Optimization

To solve the optimization problem in Eq. (3), a variable alternating optimization strategy is considered, i.e., one variable is solved while frozen the other one. In addition, the inexact augmented Lagrangian multiplier (IALM) and alternating direction method of multipliers (ADMM) are used in solving each variable, respectively. We just set reconstruction matrix ($\boldsymbol{\mathcal{Z}}$) and subspace representation ($f^{(m)}(\mathbf{X})$) as two variables. For solving the $\boldsymbol{\mathcal{Z}}$, auxiliary variable $\boldsymbol{J}$ is added. To obtain the parameters $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, stochastic sub-gradient descent method is employed. With the two updating steps for $f^{(m)}(\mathbf{X})$ and $\boldsymbol{\mathcal{Z}}$, the iterative optimization procedure of the proposed LSTN is summarized in **Algorithm 1**.

3.4 Classification

In this paper, the superiority of the proposed method is shown through the cross-domain or cross-place classification performance on the source data and target data in subspace, which can be represented as $\boldsymbol{\mathcal{X}}_S = f^{(M)}(\mathbf{X}_S)$ and $\boldsymbol{\mathcal{X}}_T = f^{(M)}(\mathbf{X}_T)$, respectively. Then, the general classifiers can be used for

Algorithm 1 The Proposed LSTN

```
Input: $\boldsymbol{X}_S$, $\boldsymbol{X}_T$ ,$\lambda$, $\gamma$.
Procedure:
1. Initialize: Add auxiliary variable $\boldsymbol{J}$, where $\boldsymbol{\mathcal{Z}} = \boldsymbol{J}$.
               Add Lag-multipliers $\boldsymbol{R}_1$ and penalty parameter $\mu$.
2. While not converge do
  2.1 Step1: Do forward propagation to all data points;
  2.2 Step2: Compute objective function;
  2.3 Step3: Fix $\boldsymbol{J}$ and $\boldsymbol{\mathcal{Z}}$, and update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ in $f^{(m)}$;
        For $m = M, M-1, ..., 1$ do
          Compute $\nabla(\mathbf{W}^{(m)})$ and $\nabla(\mathbf{b}^{(m)})$ by back-propagation operator using
          $\nabla(\mathbf{W}^{(m)}) = 2\mathbf{L}_S{}^{(m)}(\mathbf{h}_S^{(m-1)})^T - 2\mathbf{L}_T{}^{(m)}(\mathbf{h}_T^{(m-1)})^T + 2\gamma\mathbf{W}^{(m)a}$ and
          $\nabla(\mathbf{b}^{(m)}) = 2\mathbf{L}_S{}^{(m)} - 2\mathbf{L}_T{}^{(m)} + 2\gamma\mathbf{b}^{(m)a}$;
        end
        For $m = 1, 2, ..., M$ do
          Update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ according to Gradient descent operator[24];
        end
  2.4 Step4: Fix $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, and update $\boldsymbol{\mathcal{Z}}$ using ADMM;
        Fix $\boldsymbol{\mathcal{Z}}$, and update $\boldsymbol{J}$ by using the SVT operator;
        Fix $\boldsymbol{J}$, and compute $\nabla(\boldsymbol{\mathcal{Z}})$ by back-propagation operator using
          $\nabla(\boldsymbol{\mathcal{Z}}) = 2(\mathbf{h}_S^{(M)})^T(\mathbf{h}_S^{(M)}\boldsymbol{\mathcal{Z}} - \mathbf{h}_T^{(M)}) + \mathbf{R}_1 + \mu(\boldsymbol{\mathcal{Z}} - \mathbf{J})$;
        Update $\boldsymbol{\mathcal{Z}}$ according to Gradient descent operator[24];
  2.5 Update the multiplier $\boldsymbol{R}_1$ by $\boldsymbol{R}_1 = \boldsymbol{R}_1 + \mu(\boldsymbol{\mathcal{Z}} - \boldsymbol{J})$
  2.6 Update the parameter $\mu$ by $\mu = min(\mu \times 1.01, max_\mu)$
  2.7 Check convergence
end while
Output: $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ and $\boldsymbol{\mathcal{Z}}$.
```

[a] $\mathbf{L}_S{}^{(M)} = (\mathbf{h}_S^{(M)}\boldsymbol{\mathcal{Z}} - \mathbf{h}_T^{(M)})\boldsymbol{\mathcal{Z}}^T \odot \varphi'(\boldsymbol{Z}_S^{(M)})$
$\mathbf{L}_S{}^{(m)} = (\mathbf{W}^{(m+1)})^T\mathbf{L}_S{}^{(m+1)} \odot \varphi'(\boldsymbol{Z}_S^{(m)})$
$\mathbf{L}_T{}^{(M)} = (\mathbf{h}_S^{(M)}\boldsymbol{\mathcal{Z}} - \mathbf{h}_T^{(M)}) \odot \varphi'(\boldsymbol{Z}_T^{(M)})$
$\mathbf{L}_T{}^{(m)} = (\mathbf{W}^{(m+1)})^T\mathbf{L}_T{}^{(m+1)} \odot \varphi'(\boldsymbol{Z}_T^{(m)})$

training on the source data $\boldsymbol{\mathcal{X}}_S$ with label $\boldsymbol{\mathcal{Y}}_S$ in unsupervised mode. Finally, the recognition performance is verified and compared based on the target data $\boldsymbol{\mathcal{X}}_T$ and target label $\boldsymbol{\mathcal{Y}}_T$.

4 Experiments

In this section, the experiments on several benchmark datasets [7, 9, 16] have been exploited for evaluating the proposed LSTN method, including: cross-domain 4DA office data and cross-place Satellite-Scene5 (SS5) dataset [21]. Several related transfer learning methods based on feature transformation and reconstruction, such as GFK [8], SA [6], DIP [2], TJM [19], LSSA [1], CORAL [28],JDA [18], JGSA [34], ILS [10], even the latest LDADA [21] have been com-

Fig. 2: Some samples from 4DA datasets

pared and discussed. As the LSTN model we proposed can be regarded as a shallow domain adaptation approach, therefore, the shallow feature (4DA SURF features) and deep feature (4DA-VGG-M features) can be fed into the model. The deep transfer learning methods are also used to compare with our method.

Results on 4DA Office dataset (Amazon, DSLR, Webcam and Caltech 256) [8]:

This dataset is a standard cross-domain object recognition dataset. Four domains such as Amazon (A), DSLR (D), Webcam (W), and Caltech 256 (C) are included in 4DA dataset, which contains 10 object classes. With the domain adaptation setting, 12 cross-domain tasks are tested, e.g. $A \rightarrow D$, $C \rightarrow D$. In our experiment, the configuration is followed in [17] by full protocol. We compare the classification performance of LSTN using the conventional 800-bin SURF features [8]. The recognition accuracies are reported in Table 1, from which we observe that the performance of our method is higher than state-of-the-art method and 3.6% higher than the latest LDADA method in average cross-domain recognition performance.

CNN features (FC7 of VGG-M) of 4DA datasets are also used to verify the classification performance. This allows us to compare against several recently reported results. We have chosen the first nine tasks to exploit the performance in our method. Average multi-class accuracy is used as the performance measure. We have highlighted the best results in Table 2, from which we can observe that the proposed LSTN (92.2%) is better than LDADA (92.0%), and shows a superior performance over other related methods.

The compared methods above are shallow transfer learning. It is interesting to compare with deep transfer learning methods, such as AlexNet [14], DDC [29], DAN [17] and RTN [20]. The first nine tasks are used to verify the classification performance. The comparison is described in Table 3, from which we can observe that our proposed method ranks the second in average performance (92.2%), which is inferior to the residual transfer network (RTN), but still better than other deep transfer learning models. The comparison shows that the proposed LSTN, as a shallow transfer learning method, has a good competitiveness.

Table 1: Recognition accuracy (%) in 4DA-SURF features and time cost (s) in LSTN

4DA Tasks	NA	GFK	DIP	SA	JDA	TJM	LSSA	CORAL	ILS	JGSA	LDADA	Ours
$A \to D$	36.3	39.9	47.8	37.7	44.2	45.6	39.2	38.5	37.3	**49.4**	39.1	46.5(2.75s)
$C \to D$	37.6	42.9	46.9	40.9	44.1	38.4	46.3	31.8	38.3	46.0	41.5	**52.2**(3.65s)
$W \to D$	73.6	78.7	79.6	70.3	86.3	83.6	57.4	80.9	80.1	78.5	74.6	**86.6**(**0.51s**)
$A \to C$	37.3	44.3	41.4	44.8	**44.9**	42.4	40.9	37.1	35.4	40.8	38.4	44.6(24.42s)
$W \to C$	23.7	32.0	30.0	32.3	29.8	33.3	29.7	32.5	33.1	29.7	31.7	**36.8**(7.37s)
$D \to C$	25.5	30.8	29.3	31.1	34.4	32.3	31.2	27.8	**36.8**	30.2	29.9	36.2(5.06s)
$D \to A$	28.4	40.4	31.6	40.8	**44.6**	37.1	32.9	31.9	41.9	39.0	40.6	40.3(3.86s)
$W \to A$	28.7	38.3	33.8	**43.3**	42.0	39.5	38.5	39.4	38.0	34.6	35.1	40.4(5.67s)
$C \to A$	46.4	56.6	56.4	54.4	**59.8**	54.4	51.5	45.9	28.5	55.1	54.8	53.7(23.87s)
$C \to W$	39.0	48.1	51.2	45.8	50.1	44.0	43.9	37.8	28.4	49.7	**60.2**	47.5(5.53s)
$D \to W$	61.6	80.3	67.5	74.4	83.3	83.7	42.6	69.4	81.5	75.1	74.7	**86.1**(0.63s)
$A \to W$	34.4	42.7	44.8	44.1	47.0	39.5	40.2	37.9	40.0	**59.0**	49.3	42.7(4.33s)
Average	39.4	47.9	46.7	46.7	50.9	47.8	41.2	42.6	43.3	48.9	47.5	**51.1**

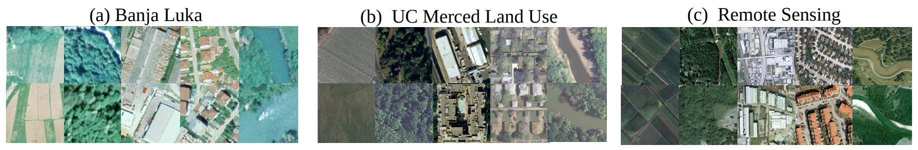


Fig. 3: Some samples from Satellite-Scene5 datasets

Notably, the 4DA and CNN features in 4DA tasks are challenging benchmarks, which attract many competitive approaches for evaluation and comparison. Therefore, excellent baselines have been achieved.

Results on SS5 (Satellite-Scene5)(Banja Luka (B), UC Merced Land Use (U), and Remote Sensing (R)):

To validate that LSTN is general and can be applied to other images with different characteristics and particularly to the categories which are not included in the ImageNet dataset, we conduct evaluations on a cross-place satellite scene dataset. Three publicly available datasets as Banja Luka (B) [23], UC Merced Land Use (U) [33], and Remote Sensing (R) [3] datasets are selected specifically. In experiment, for cross-place classification, 5 common semantic classes: farmland/field, trees/forest, industry, residential, and river have been explored, respectively. There are 6 DA problem settings on this dataset. Several example images are shown in Fig. 3.

We follow the full protocol explained in LDADA [21], which allows us to compare against several recently reported results on the SS5 dataset. Results are shown in Table 4. We observe that LSTN (76.4%) which equals to the recent ILS [10] and LDADA [21], still outperforms other competitors and consistently improves the cross-place accuracy in DA tasks. The result suggests that LSTN should also be applicable to other general visual recognition problems.

Table 2: Recognition accuracy (%) in 4DA-VGG-M model with shallow methods

4DAVGG	$A \to D$	$C \to D$	$W \to D$	$A \to C$	$W \to C$	$D \to C$	$D \to A$	$W \to A$	$C \to A$	Average
NA	77.2	89.9	99.0	81.7	77.3	75.0	83.6	85.5	91.5	84.5
GFK	85.5	91.0	98.0	85.3	81.3	82.3	90.8	90.2	93.6	88.7
DIP	83.4	91.4	98.0	86.0	81.2	81.0	90.0	88.4	93.3	88.1
SA	89.6	**95.0**	98.0	77.1	77.9	78.6	83.8	87.3	93.2	86.7
JDA	91.3	93.2	96.1	**90.1**	86.7	84.8	91.7	93.8	93.7	91.3
TJM	89.9	90.8	97.6	86.4	81.4	81.8	91.4	91.1	93.9	89.4
LSSA	86.2	91.8	95.3	88.0	82.8	81.7	91.2	91.8	93.8	89.2
CORAL	76.2	87.6	98.0	80.1	77.6	73.1	84.5	90.7	91.6	84.4
ILS	83.7	87.7	96.9	86.2	87.0	85.7	91.2	93.6	93.1	89.5
JGSA	**94.1**	94.4	96.1	87.2	82.3	85.2	93.8	**94.9**	94.2	91.4
LDADA	90.0	93.2	99.6	88.7	**88.3**	84.8	**94.2**	94.3	**95.1**	92.0
Ours	91.1	93.0	**100.0**	89.6	88.1	**87.8**	93.4	92.9	94.1	**92.2**

Table 3: Recognition accuracy (%) in 4DA-VGG-M model with deep methods

4DAVGG	$A \to D$	$C \to D$	$W \to D$	$A \to C$	$W \to C$	$D \to C$	$D \to A$	$W \to A$	$C \to A$	Average
AlexNet	88.3	89.1	100.0	84.0	77.9	81.0	89.0	83.1	91.3	87.1
DDC	89.0	88.1	100.0	85.0	78.0	81.1	89.5	84.9	91.9	87.5
DAN	92.4	90.5	100.0	85.1	84.3	82.4	92.0	92.1	92.0	90.1
RTN	**94.6**	92.9	100.0	88.5	**88.4**	84.3	**95.5**	**93.1**	**94.4**	**92.4**
Ours	91.1	**93.0**	100.0	**89.6**	88.1	**87.8**	93.4	92.9	94.1	92.2

5 Discussion

5.1 Parameter Setting

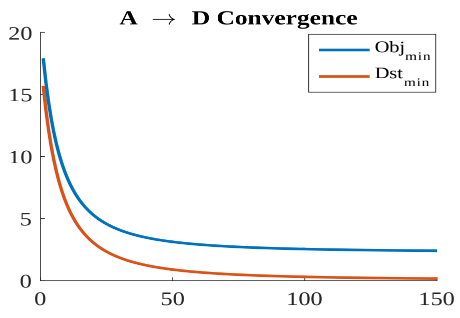
In our method, two trade-off coefficients γ and λ are involved. γ and λ are fixed as 0.01 and 1 in experiments, respectively. The number of iterations $T = 10$ is enough in the experiments. The Gaussian kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ =exp$(- \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2 /2\sigma^2)$ is used, where σ can be set as $\sigma = 1.4$ in the tasks. The least square classifier is used in DA experiments. In the MLP network, the number of layers is set as $M = 2$. The dimension of output in the latent space is the same as input. Tanh activation function $\varphi(\cdot)$ is adopted in MLP network. The parameters of the weights and bias are auto updated by gradient descent based on back-propagation algorithm.

5.2 Computational Complexity

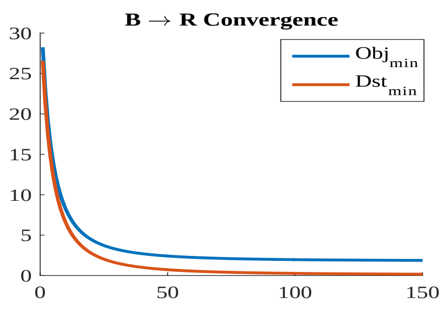
In this section, the computational complexity of the **Algorithm 1** is presented. The algorithm includes three basic steps: update $\boldsymbol{\mathcal{Z}}$, update $\boldsymbol{J}$, and update $f^{(m)}$. The computation of $f^{(m)}$ involves $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, and the complexity is

Table 4: Recognition accuracy (%) in SS5 Setting and time cost (s) in LSTN

SS5 Tasks	NA	GFK	DIP	SA	JDA	TJM	LSSA	CORAL	ILS	JGSA	LDADA	Ours
$B \to R$	46.4	56.9	35.9	46.3	37.0	46.1	44.9	59.2	**75.3**	51.4	51.8	59.8(2.01s)
$R \to B$	39.4	48.5	41.9	46.9	65.6	52.1	60.7	27.3	56.1	25.1	58.9	**66.3**(2.43s)
$B \to U$	60.8	66.2	56.8	57.2	61.0	56.4	64.8	57.6	78.9	59.4	**81.6**	76.8(4.48s)
$U \to B$	69.9	64.4	61.4	62.8	74.5	63.4	57.5	66.6	66.2	**77.3**	70.5	75.8(4.53s)
$R \to U$	72.2	88.6	76.2	79.6	91.6	85.0	89.6	80.4	95.4	94.6	**97.2**	90.8(2.02s)
$U \to R$	75.2	80.7	69.9	72.3	95.1	79.6	83.5	71.2	86.7	97.4	**98.1**	89.1(1.63s)
Average	60.7	67.6	57.0	60.9	70.8	63.8	66.8	60.4	**76.4**	67.5	**76.4**	**76.4**



(a)$A \to D$ task in 4DA



(b)$B \to R$ task in SS5

Fig. 4: Convergence analysis on different tasks of LSTN model

$O(2MN^2)$. The computation of updating $\boldsymbol{J}$ and $\boldsymbol{\mathcal{Z}}$ is $O(N^2)$. Suppose that the number of iterations is T, then the total computational complexity of LSTN can be expressed as $O(T \times 2MN^2) + O(TN^2)$.

In LSTN model, CPU is enough for model optimization, without using GPU. The time cost is much lower as shown in the last column of Table 1 and Table 4. All experiments are implemented on the computer with Intel i7-4790K CPU, 4.00GHz, and 16GB RAM. The time cost is calculated under this setting. It is noteworthy that the time of data preprocessing and classification is excluded.

5.3 Convergence

In this section, the convergence will be discussed. We have conducted the experiments on 4DA ($A \to D$) and SS5 ($B \to R$), respectively. The convergence of our LSTN method is explored by observing the variation of the objective function. In the experiments, the number of iterations is set to be 150 for verification the convergence better. The variation of the objective function (Obj_{min}) and reconstruction loss function (Dst_{min}) are described in Fig. 4. It is clear that the objective function and reconstruction loss function decrease to a constant value after several iterations. By running the algorithm, on 4DA and SS5 tasks, respectively, we can observe the good convergence of LSTN.

6 Conclusion

In this paper, we show that one can achieve the effect of DA by combining both advantages of subspace learning and neural network. Specifically, a reconstruction-based transfer learning approach called LSTN is proposed. It offers a simple but effective solution for DA with ample scope for improvement. In the method, we embed features/pixels of source and target into reproducing kernel Hilbert space (RKHS), in which the high dimensional features are mapped to nonlinear latent subspace by feeding them into MLP network. Leveraging the simple MLP, not only the layers can be optimized directly to avoid a pre-trained model which needs large-scale data, but also the adaptation behaviors can be achieved by joint learning a set of hierarchical nonlinear subspace representation and optimal reconstruction matrix simultaneously. Extensive experiments are conducted to justify our proposition in both effectiveness and efficiency. Results demonstrate that LSTN is applicable to small sample sizes, outperforms existing non-deep DA approaches, exhibits comparable accuracy against recent deep DA alternatives.

References

1. Aljundi, R., Emonet, R., Muselet, D., Sebban, M.: Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. CVPR (2015)
2. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV. pp. 769–776 (2013)
3. Dai, D., Yang, W.: Satellite image classification via two-layer sparse coding with biased image representation. IEEE Geoscience & Remote Sensing Letters **8**(1), 173–176 (2011)
4. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf:a deep convolutional activation feature for generic visual recognition. In: PMLR. pp. 647–655 (2014)
5. Duan, L., Xu, D., Tsang, I.W., Luo, J.: Visual event recognition in videos by learning from web data. In: CVPR. pp. 1959–1966 (2010)
6. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV. pp. 2960–2967 (2014)
7. Gaidon, A., Zen, G., Rodriguez-Serrano, J.A.: Self-learning camera: Autonomous adaptation of object detectors to unlabeled video streams. arXiv (2014)
8. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR. pp. 2066–2073 (2012)
9. Gong, B., Grauman, K., Sha, F.: Learning kernels for unsupervised domain adaptation with applications to visual object recognition. IJCV **109**(1-2), 3–27 (2014)
10. Herath, S., Harandi, M., Porikli, F.: Learning an invariant hilbert space for domain adaptation. CVPR pp. 3956–3965 (2017)
11. Hoffman, J., Tzeng, E., Park, T., Zhu, J.Y., Isola, P., Saenko, K., A.Efros, A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
12. Hu, J., Lu, J., Tan, Y.P.: Deep transfer metric learning. In: CVPR. pp. 325–333 (2015)
13. Jhuo, I.H., Liu, D., Lee, D., Chang, S.F.: Robust visual domain adaptation with low-rank reconstruction. In: CVPR. pp. 2168–2175 (2012)

14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NIPS **25**(2), 1097–1105 (2012)
15. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR. pp. 1785–1792 (2011)
16. Liu, D., Hua, G., Chen, T.: A hierarchical visual model for video object summarization. IEEE Trans. PAMI **32**(12), 2178–2190 (2010)
17. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML. pp. 97–105 (2015)
18. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: ICCV. pp. 2200–2207 (2014)
19. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer joint matching for unsupervised domain adaptation. In: CVPR. pp. 1410–1417 (2014)
20. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: NIPS. pp. 136–144 (2016)
21. Lu, H., Shen, C., Cao, Z., Xiao, Y., Hengel, A.V.D.: An embarrassingly simple approach to visual domain adaptation. IEEE Transactions on Image Processing **PP**(99), 1–1 (2018)
22. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR. pp. 1717–1724 (2014)
23. Risojevic, V., Babic, Z.: Aerial image classification using structural texture similarity **19**(5), 190–195 (2011)
24. Rosasco, L., Verri, A., Santoro, M., Mosci, S., Villa, S.: Iterative projection methods for structured sparsity regularization. Computation (2009)
25. Shao, M., Kit, D., Fu, Y.: Generalized transfer subspace learning through low-rank constraint. IJCV **109**(1-2), 74–93 (2014)
26. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: CVPR. pp. 806–813 (2014)
27. Simon, K., Jonathon, S., Le, Q.V.: Do better imagenet models transfer better?. arXiv preprint arXiv:1805.08974v1 (2018)
28. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: AAAI. vol. 6, p. 8 (2016)
29. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV. pp. 4068–4076 (2015)
30. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation (2017), cVPR
31. Wang, S., Zhang, L., Zuo, W.: Class-specific reconstruction transfer learning via sparse low-rank constraint. In: ICCVW. pp. 949–957 (2017)
32. Xie, M., Jean, N., Burke, M., Lobell, D., Ermon, S.: Transfer learning from deep features for remote sensing and poverty mapping. arXiv (2015)
33. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Sigspatial International Conference on Advances in Geographic Information Systems. pp. 270–279 (2010)
34. Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. CVPR pp. 5150–5158 (2017)
35. Zhang, L., Zhang, D.: Robust visual knowledge transfer via extreme learning machine-based domain adpatation. IEEE Trans. Image Processing **25**(3), 4959–4973 (2016)
36. Zhang, L., Zuo, W., Zhang, D.: Lsdt: Latent sparse domain transfer learning for visual adaptation. IEEE Trans Image Processing **25**(3), 1177–1191 (2016)